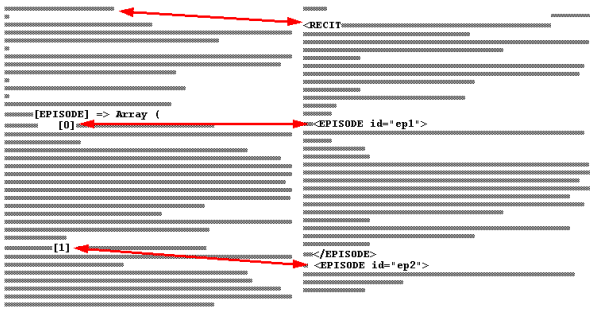
# 4. Parsing avec "Simple XML"

### Résumé des fonctionalités

- La fonction "simplexml_load_file" permet de parser un document XML dans une structure PHP qui ressemble à des indexed arrays dans des indexed arrays. On peut donc facilement accéder à des éléments (avec des sélecteurs de type "array").
- On peut aussi effectuer des recherches avec une expression XPath.
- Du manuel: The SimpleXML extension provides a very simple and easily usable toolset to convert XML to an object that can be processed with normal property selectors and array iterators
- *Avantage* : On peut très facilement intégrer une structure XML complète dans un programme PHP et ensuite la manipuler.
- *Désavantage* :
  - Solution particulière à PHP, autrement dit ce type d'approche ne se retrouve pas dans d'autres langages de programmation (contrairement à DOM et SAX).
  - Le nom de l'élément racine disparaît dans la nature (!)
- Note: En PHP 4.x une fonctionalité similaire était disponible sous le nom "xmltree")

### Principe illustré avec un exemple:




Un élément avec enfants devient un simple XML Object
**Ses enfants sont accessibles par une clef (nom de la balise)**
La clef retourne soit un contenu, un XML Object, une liste de XML Objects

### Exemple 4-1: Lire un fichier XML avec SimpleXML

- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/simplexml0.php
- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/ (répertoire)

```
<?php
  if (file_exists('story.xml')) {
  $xml = simplexml_load_file
('story.xml');

  echo "<hr>Here is a dump of the data structure:";

  echo "<pre>";
  print_r($xml);
  echo "</pre>";

} else {
  exit('Failed to open story.xml.');
}
?>
```

- `simplexml_load_file()` charge un fichier XML
- `print_r()` imprime une structure de données de façon "human-readable"
- `var_dump()` imprime plus de détails "techniques".
- Donc faites un "print_r" avant de se lancer dans programmation de l'extration des données qui vous intéressent !!

### Exemple 4-2: Extraire des données avec SimpleXML

- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/simplexml1.php
- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/ (répertoire)

```
$xml = simplexml_load_file('story.xml');

echo "<hr>Here we just display some elements (i.e. &lt;SousBut&gt; and &lt;Resultat&gt; elements found in RECIT->FIL->EPISODE). &lt;FIL&gt; can contain several &lt;EPISODE&gt;";

  $episodes = ($xml->FIL->EPISODE
);
  foreach ($episodes as $episode) {
    echo "<p>Episode:</p> ";
    echo"<pre>";
    printf("Sousbut: %s\n", $episode->SousBut);
    printf("Resultat: %s\n", $episode->Resultat);
    print "----\n";
    echo"</pre>";
```

- `$xml->FIL->EPISODE` collectionne tous les eléments "EPISODE" filles de "FIL".
- foreach `($episodes as $episode)` ... est une technique standarde pour boucler sur tous les éléments d'un array. `$episode` sera lié à chaque item trouvé lors d'un passage.

- `$episode->SousBut` extrait l'élément SousBut

**Exemple 4-3: Extraire des données avec SimpleXML et Xpath**

- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/simplexml2.php
- http://tecfa.unige.ch/guides/php/examples/simplexml-functions/ (répertoire)
- Meme exemple que le précédant sauf qu'on utilise XPath pour extraire la liste des épisodes.

```
$xml = simplexml_load_file('story.xml');


$episodes = $xml->xpath('//EPISODE');
```

- $xml->xpath('//EPISODE') collectionne tous les eléments "EPISODE" filles de "FIL".
- foreach `($episodes as $episode)` ... est une technique standarde pour boucler sur tous les éléments d'un array. `$episode` est un item.

-- TIE